

# 라즈베리-파이에서 USB 센서의 Plug&Play 기법

은성배<sup>1,\*</sup> · 소선섭<sup>2</sup>

<sup>1</sup>한남대학교 · <sup>2</sup>공주대학교

## A Plug&Play Scheme of Usb Sensros In Raspberry-Pi

Seongbae Eun<sup>1,\*</sup> · Sun Sup So<sup>2</sup>

<sup>1</sup>Hannam University · <sup>2</sup>Kongju National University

E-mail : sbeun@hnu.kr / triples@kongju.ac.kr

### 요 약

IoT 장치를 개발하는 방법은 응용이 요구하는 센서를 아두이노나 라즈베리 파이 같은 플랫폼에 장착하고 그 센서의 드라이버와 응용을 작성하는 것이다. 이때 그 센서를 위한 구동 드라이버가 이미 작성되어 있고 응용에서 그 드라이버를 표준화된 API로 접근할 수 있다면 센서의 Plug&Play가 가능할 것이다. 이를 위한 과거의 방식은 센서 인터페이스가 너무 복잡하여 현재의 플랫폼에서는 사용하기 어렵다. 본 논문에서는 USB단자를 갖는 표준화된 센서와 드라이버가 라즈베리 파이에 Plug되면 자동으로 드라이버가 설치되는 기법을 제안한다. 응용 개발자는 센서 드라이버는 고민하지 않고 Linux의 파일 접근 API로 센서 값을 얻을 수 있다. 제안된 기법은 현재 라즈베리-파이 상에서 구현 중이다.

### ABSTRACT

The way to develop an IoT device is to mount the sensor required by the application on a platform such as Arduino or Raspberry Pi, and write the sensor driver and application. At this time, if the driving driver for the sensor has already been written and the application can access the driver as a standardized API, then Plug&Play of the sensor will be possible. The old way to do this is because the sensor interface is too complicated to use on the current platform. In this paper, when a standardized sensor and driver with a USB terminal are plugged into the Raspberry Pi, we propose a method for automatically installing the driver. Application developers can get sensor values through Linux's file access API without worrying about sensor drivers. The proposed technique is currently being implemented on Raspberry-Pi.

### 키워드

Plug&Play, Sensors, Standardized, Raspberry-Pi, Linux

### 1. 서 론

IoT 장치를 개발하는 방법은 응용이 요구하는 센서를 아두이노나 라즈베리 파이 같은 플랫폼에 장착하고 그 센서의 드라이버와 응용을 작성하는 것이다. 이때, 센서의 드라이버를 작성하는 것은 센서 하드웨어를 이해해야 하므로 소프트웨어 개발자에게는 어려운 일이다.

Linux[1] 등에서처럼 I/O 장치의 드라이버를 장

치 제조사가 개발하고 응용 프로그래머는 표준화된 장치 접근 API, 즉, open(), read(), write(), close() 등으로 접근한다면 응용 개발의 효율이 높아질 것이다.

본 연구팀은 과거에 이를 위한 센서 추상화 기법 [2]을 연구하였는데 이를 통하여 센서 노드 개발을 용이하게 할 수 있었다. 그 기법은 센서의 다양성 극복을 위하여 10핀을 갖는 표준 센서 하드웨어를 제안했는데 너무 복잡하다는 문제를 갖는다.

본 논문에서는 현재 IoT 장치 개발에 많이 사용되는 라즈베리-파이 등의 IoT 개발 플랫폼을 고려

\* corresponding author

하여 USB 인터페이스를 갖는 센서 추상화 기법을 제안한다.

## II. 배경

### 2.1. 기존의 USN 응용 개발

기존의 USN 응용 개발 방법을 그림 1 에서 도식한다. 그림에서 응용 개발자는 사용자의 응용 개발 요구에 따라 센서와 MCU, RF 모듈등을 선정, HW를 조립한다. SW 개발은 Tiny-OS[2-5], SoS[6], Mantis[7], Nano-Q+[8] 등을 활용하여 개발하거나 운영체제 도움없이 FW를 직접 개발한다.

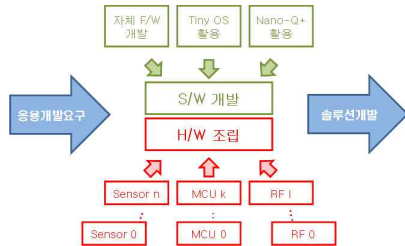


그림 1. 기존 센서 노드 개발법[2]

이러한 개발 구조에서는 응용 개발자가 HW, SW, 디바이스 드라이버 등을 모두 제작해야 한다. 응용 개발자가 자체 보유한 플랫폼은 응용에 종속적이므로 수시로 제작되고 수정돼야 하므로 규모의 경제를 이루지 못한다. 개발 가격이 비싸지고 이 때문에 응용 개발 요구가 늘어나기 어렵다. 응용 개발이 많아질 수록 개발 가격이 떨어지고 그것이 응용 개발 요구를 확대하는 선순환 구조를 이루기가 어렵다.

### 2.2. IEEE1451

IEEE1451[9]은 미국의 NIST에서 주도하고 있는 표준으로서 1451.0 ~ 1451.6, 총 7개의 표준으로 나뉘어 진행 중이다. 이 표준은 트랜스듀서의 다양성을 극복하는데 초점을 맞추고 있다.

IEEE1451에서 정의하는 트랜스듀서와 NCAP과의 인터페이스는 센서노드 플랫폼을 개발할 때 많이 참고할 수 있으나 1451표준에서는 전원이 별도로 공급되고 저전력을 위한 제어도 어려우므로 1451 표준이 센서노드 플랫폼을 대체한다고 볼 수 없다.

### 2.3. 센서 추상화 기법

본 연구팀은 과거에 센서 추상화 기법[2]을 제시하여 센서의 Plug&Play를 지원하였다. 센서 추상화 기법은 표준화된 센서노드 플랫폼 기반의 USN 응용개발 방법을 제공한다. 응용 개발자는 플랫폼이 제공하는 표준화된 API 위에서 응용프로그램을 개발하면 된다. 센서 디바이스 공급자는 표준화된 HW, SW 인터페이스에 맞게 디바이스 드라이버를 개발, 공급하면 된다. 이를 위하여 센서노드 플랫폼

이 가져야 할 기본적인 구조를 제시한하였다. HW가 갖추어야 할 기본 구조를 제시하고 센서 디바이스를 접근할 때 사용할 수 있는 표준화된 API를 제시하였고 디바이스 드라이버가 OS와 접속되는 방안을 제시하였다.

이 방식의 문제는 센서 하드웨어 개발자가 10핀의 센서 커넥터를 사용한다는 점인데 이는 최근의 아두이노와 라즈베리 파이 등의 개발 플랫폼에서 지원하지 않는 것이다.

## III. USB 센서 기반 Plug&Play

### 3.1. 시스템 구성

그림 2는 전체 시스템의 구조를 보여준다.

#### 1) 응용개발자

센서 응용을 개발할 때 과거와 달리 센서 자체의 디바이스 드라이버를 작성할 필요가 없이 리눅스의 open(), close(), 센서의 경우엔 read(), 구동기의 경우엔 write()를 이용하여 응용을 작성하면 된다. 이는 센서가 변하더라도 응용이 변할 필요가 없다는 것이다. 예를 들어 A라는 온도 센서에서 B로 변경되더라도 응용이 변할 필요가 없다.

#### 2) 플랫폼

라즈베리-파이나 아두이노, 기타 ATmega128 기반 노드 등 다양하나 USB 포트를 지원해야 한다. 라즈베리-파이의 경우에는 Linux가 동작하므로 단순하나 아두이노나 FW 기반으로 동작하는 플랫폼에서는 특별한 기능이 필요하다.

#### 3) USB 센서/구동기 스틱

플랫폼에 부착되는 센서 스틱으로서 USB를 통신 포트로 사용하며 전원을 공급할 수 있다는 점에서 센서 개발에 도움이 된다.

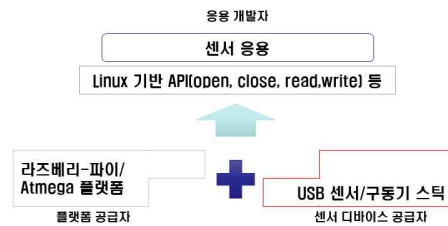


그림 2. 전체 시스템 구성도

### 3.2. USB 센서 스틱의 구조

그림 3은 센서 스틱의 구조를 보여준다. 그림에서 센서 모듈은 USB 포트에 장착할 수 있는 스틱 구조를 갖는다. 그림에서 보듯이 센서 모듈 내부는 1) USB 인터페이스를 관리하는 USB 관리 모듈, 2) 다양한 센서가 부착되는 센서 H/W, 3) 센서 디바이스 드라이버가 저장된 센서 D/D로 구성된다.

#### 1) USB 관리 모듈

이 모듈은 USB 통신을 지원함과 동시에 센서에 전원을 공급하는 기능, USB가 부착되는 순간 라지

베리-파이의 USB 드라이버와 연동하여 센서 디바이스 드라이버를 업로딩하는 기능, 센서로부터 얻은 데이터를 USB 직렬 통신을 이용하여 전송하는 기능 등을 제공한다.

#### 2) 센서 H/W

이는 센서 그 자체이며 온도 센서, 습도 센서 등의 HW를 의미한다.

#### 3) 센서 D/D

이는 장착된 센서에서 데이터를 얻을 수 있도록 지원하는 디바이스 드라이버이며 리눅스 디바이스 드라이버 제작법에 따라 작성된다. USB 센서 모듈이 부착되면 리눅스 내부의 USB 디바이스 드라이버에서 정해진 절차에 따라 센서 D/D가 리눅스로 전송된다. 그 이후 센서 D/D로서 동작한다.

USB A/B/C Type

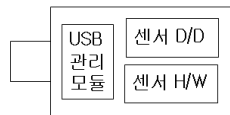


그림 3. USB 센서 스틱 구조도

## IV. 결 론

본 논문에서는 USB 단자를 갖는 표준화된 센서 및 구동기 스틱이 라즈베리-파이에 부착될 때 자동으로 디바이스 드라이버가 업로드되어 동작하는 센서 Plug&Play 기법을 제안하였다. 응용 개발자는 센서 드라이버는 고민하지 않고 Linux의 파일 접근 API로 센서 값을 얻을 수 있다. 현재 라즈베리-파이의 Linux 운영체제와 연동하도록 구현 중이다.

## References

- [1] A. Rubini, Linux Device Drivers, O'Reilly & Associates, Inc., 1998.
- [2] M. Yang, S. So, S. Eun, B. Kim, and J. Kim, "Sensos: A Sensor Node Operating System with a Device Management Scheme for Sensor Nodes," International Conference on Information Technology (ITNG'07), 2007, pp. 134-139.
- [3] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, A. Woo, E. Brewer, and D. Culler, "The emergence of networking abstractions and techniques in tinyos," Proc. of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2004), 2004.
- [4] T. Schmid, H. Dubois-Ferriere, and M. Vetterli, "Sensorscope: experiences with a wireless building monitoring sensor network," Proc. of Workshop on Real-World Wireless Sensor Networks, 2005
- [5] P. Volgyesi and A. Ledeczi, "Component-based development of networked embedded applications," Proc. of 28<sup>th</sup> Euromicro Conference, 2002.
- [6] C. C. Han, R. Kumar, R. Shea, E. Kohler, and M.B. Srivastava, "A dynamic operating system for sensor nodes," Proc. of MobiSys, 2005, pp.163-176.
- [7] H. Abrach, S. Bhatti, J. Carlson, H. Dai, J. Rose, A. Sheth, B. Shucker, J. Deng, and R. Han, "MANTIS: System Support For Multimodal NeTworks of In-situ Sensors," Proc. of 2<sup>nd</sup> ACM International Workshop on Wireless Sensor Networks and Applications, 2003, pp.50-59.
- [8] S. Park, J. Kim, K. Lee, K. Shin, and D. Kim, "Embedded Sensor Networked Operating System," Proc. of 9<sup>th</sup> IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing, 2006.
- [9] Institute of Electrical and Electronics Engineers, Inc., "IEEE Standard for Smart Transducer Interface for Sensors and Actuators - Network Capable Application Processor (NCAP) Information Model," Mixed-Mobile Communication Working Group of the Technical Committee on Sensor Technology TC-9 of the IEEE Instrumentation and Measurement Society, June 1999.